

NASA Technical Memorandum 86020

NASA-TM-86020 19850010320

On the Connection Between Multigrid and Cyclic Reduction

Marshal L. Merriam

September 1984

LIBRARY COPY

FEB 2 1985

LANGLEY RESEARCH CENTER
LIBRARY, NASA
HAMPTON, VIRGINIA



National Aeronautics and
Space Administration



NF00863

On the Connection Between Multigrid and Cyclic Reduction

Marshal L. Merriam, Ames Research Center, Moffett Field, California



National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

1185-18629[#]

ON THE CONNECTION BETWEEN MULTIGRID AND CYCLIC REDUCTION

Marshal L. Merriam

NASA Ames Research Center
Moffett Field, Ca 94035

Abstract

A technique is shown whereby it is possible to relate a particular multigrid process to cyclic reduction using purely mathematical arguments. This technique suggests methods for solving Poisson's equation in 1-, 2-, or 3-dimensions with Dirichlet or Neumann boundary conditions. In one dimension the method is exact and, in fact, reduces to cyclic reduction. This provides a valuable reference point for understanding multigrid techniques. The particular multigrid process analyzed is referred to here as Approximate Cyclic Reduction (ACR) and is one of a class known as Multigrid Reduction methods in the literature. It involves one approximation with a known error term. It is possible to relate the error term in this approximation with certain eigenvector components of the error. These are sharply reduced in amplitude by classical relaxation techniques. The approximation can thus be made a very good one.

1 Introduction

In the last decade a new class of relaxation schemes known as multigrid methods have appeared in the literature. These schemes solve large, sparse, wide-banded linear equations. They have many potential benefits including remarkable convergence speed which is such that the number of operations required for a solution is proportional to the number of unknowns (ref. 1).

Multigrid techniques appear to be ideal for the solution of many equations encountered in computational fluid dynamics today. In addition to their speed they have a potential for simpler implementation of component and solution adaptive grids as well as a number of other advantages detailed in (ref. 1) and (ref. 2). Although widely experimented with, they do not appear to be in widespread use for practical problems. The notable exception is the work of Antony Jameson (ref. 3). The purpose of this paper is to provide a clear and coherent explanation of a particular multigrid strategy when applied to a model problem and to show that multigrid can be thought of as an approximation to cyclic reduction.

Multigrid strategies can most easily be analyzed when they are used to solve Poisson's equation. This is because the matrix which approximates the Laplacian operator has analytically known eigenvectors and eigenvalues. Since the eigenvectors are sine functions, a Von Neumann stability analysis will usually agree quite well with an analysis which uses the eigenvectors of the Laplacian as error components. One exception occurs in the neighborhood of a boundary where the periodicity assumptions in a Von Neumann analysis break down.

It is possible in one dimension to show a link between cyclic reduction and multigrid. Although the analogy does not carry over exactly in two dimensions, it does provide some insight. The analysis extends to two and three dimensions, and allows at least Neumann and Dirichlet boundary conditions.

Von Neumann stability analysis will usually agree quite well with an analysis which uses the eigenvectors of the Laplacian as error components. One exception occurs in the neighborhood of a boundary where the periodicity assumptions in a Von Neumann analysis break down.

It is possible in one dimension to show a link between cyclic reduction and multigrid. Although the analogy does not carry over exactly in two dimensions, it does provide some insight. The analysis extends to two and three dimensions, and allows at least Neumann and Dirichlet boundary conditions.

Results are presented comparing the convergence rate of ACR with that of other classical methods. While the convergence of ACR is quite good compared to some methods its principal value lies in the insights it provides.

2 General Idea

We assume some suitable discretization of the locally linearized governing equations. This results in the system of linear equations

$$A_f \phi_f = f_f \quad (1)$$

where A_f is a matrix, ϕ_f is a vector of unknowns, and f_f is a vector containing boundary conditions and a forcing function.

With multigrid techniques as with cyclic reduction the idea is to deduce the solution to equation (1) from the solution of a simpler equation

$$A_c \phi_c = f_c \quad (2)$$

Cyclic reduction is able to exactly solve for ϕ_f from ϕ_c but multigrid can do this only in an approximate sense. Traditionally, each element of ϕ_c is approximately or exactly equal to a particular element of ϕ_f . Also, for multigrid methods, A_c is usually chosen to be of the same form as A_f in some sense, which allows the solution of equation (2) to be derived from that of a still simpler equation. In this paper we choose to examine only those schemes where A_c and ϕ_c are defined in this way.

Since both A_c and ϕ_c have been chosen there must be a unique value of f_c for which equation (2) is satisfied. This paper will address the problem of how to find f_c from the given information, A_f , A_c , and f_f .

We begin by reordering the scalar equations and unknowns in equation (1) and performing the appropriate row and column permutations on A_f .

$$\left[\begin{array}{c|c} A_1 & A_2 \\ \hline A_3 & A_4 \end{array} \right] \times \left[\begin{array}{c} \phi_e \\ \hline \phi_o \end{array} \right] = \left[\begin{array}{c} f_e \\ \hline f_o \end{array} \right] \quad (3)$$

This can be written as a set of two matrix equations with two vector unknowns

$$\begin{aligned} A_1 \phi_e + A_2 \phi_o &= f_e \\ A_3 \phi_e + A_4 \phi_o &= f_o \end{aligned} \quad (4)$$

If we solve equation (5) for ϕ_o we get

$$\phi_o = A_4^{-1}(f_o - A_3 \phi_e) \quad (6)$$

Substituting this into equation (4) gives

$$(A_1 - A_2 A_4^{-1} A_3) \phi_e = f_e - A_2 A_4^{-1} f_o \quad (7)$$

Notice that this has the form of equation (2) where

$$A_c = (A_1 - A_2 A_4^{-1} A_3) \quad (8a)$$

$$\phi_c = \phi_e \quad (8b)$$

$$f_c = f_e - A_2 A_4^{-1} f_o \quad (8c)$$

By construction ϕ_c is a subset of ϕ_f . In general f_c is not f_e . The nature of A_c will depend on the nature of A_f and on which unknowns we choose to call ϕ_e . We would like A_c and A_f to be identical discretizations of the same PDE on different sized meshes. We will show that this is possible for Poisson's equation

3 One-Dimensional Example

We will now focus on the particular set of equations which arise from discretizing Poisson's equation with the standard central differencing. In one dimension this is, for Dirichlet boundary conditions

$$\begin{bmatrix} -2 & 1 & & \\ 1 & -2 & 1 & \\ & & & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix} \times \begin{bmatrix} \phi_1 \\ \phi_2 \\ \\ \phi_{N-1} \\ \phi_N \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \\ f_{N-1} \\ f_N \end{bmatrix} \quad (9)$$

Here we have absorbed the boundary condition information into f_1 and f_N

In this case an elegant choice is to let ϕ_e be the even numbered dependent variables. When this is done equation (3) becomes

$$\left[\begin{array}{ccc|ccc} -2 & & & 1 & 1 & \\ & -2 & & & 1 & 1 \\ & & & & & \\ & & -2 & & 1 & 1 \\ & & & -2 & & \\ \hline 1 & & & -2 & & \\ 1 & 1 & & & -2 & \\ & 1 & 1 & & & -2 \\ & & & 1 & 1 & \\ & & & & 1 & \end{array} \right] \times \begin{bmatrix} \phi_2 \\ \phi_4 \\ \\ \phi_{N-3} \\ \phi_{N-1} \\ \phi_1 \\ \phi_3 \\ \phi_5 \\ \\ \phi_{N-2} \\ \phi_N \end{bmatrix} = \begin{bmatrix} f_2 \\ f_4 \\ \\ f_{N-3} \\ f_{N-1} \\ f_1 \\ f_3 \\ f_5 \\ \\ f_{N-2} \\ f_N \end{bmatrix} \quad (10)$$

For this example A_4^{-1} is just $-I/2$ where I is the identity matrix. Using this and multiplying by $2I$, equation (7) becomes

$$\begin{bmatrix} -2 & 1 & & \\ 1 & -2 & 1 & \\ & & & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix} \times \begin{bmatrix} \phi_2 \\ \phi_4 \\ \\ \phi_{N-3} \\ \phi_{N-1} \end{bmatrix} = \begin{bmatrix} f_1 & + & 2f_2 & + & f_3 \\ f_3 & + & 2f_4 & + & f_5 \\ \\ f_{N-4} & + & 2f_{N-3} & + & f_{N-2} \\ f_{N-2} & + & 2f_{N-1} & + & f_N \end{bmatrix} \quad (11)$$

which is the desired result. Note that A_e in equation (11) has the same tridiagonal structure that A_f had in equation (9) and therefore can be reduced in the same way. If N is one less than a power of two the reduction process can be continued recursively until only one

equation remains. Once ϕ_e is known ϕ_o can be found by direct application of equation (6). This process is well known and is one of a class known as cyclic reduction (ref. 4). It also represents a multigrid process where the restriction, sometimes known as the fine to coarse interpolation, is just

$$(f_c)_i = (f_f)_{2i-1} + 2(f_f)_{2i} + (f_f)_{2i+1} \quad (12)$$

and the coarse to fine grid interpolation stencil is just

$$(\phi_f)_{2i} = (\phi_c)_i \quad (13a)$$

$$(\phi_f)_{2i+1} = \frac{1}{2} \left[(\phi_c)_i + (\phi_c)_{i+1} - (f_f)_{2i+1} \right] \quad (13b)$$

where the required values at 0 and $(N+1)/2$ are

$$(\phi_c)_o = (\phi_c)_{(N+1)/2} = 0 \quad (13c)$$

In this way the restriction and interpolation are accomplished using the original difference equations. By using the analysis techniques of cyclic reduction we are able to find an exact interpolation and restriction in one dimension.

4 Two Dimensions (Restriction and Interpolation)

In two dimensions the multigrid processes defined by Brandt and others depart from standard cyclic reduction for the case of Poisson's equation on a rectangle. This comes mostly from the choice of ϕ_e .

If we index the unknowns as ϕ_{ij} , corresponding to their x and y locations on the computational mesh, we see that cyclic reduction chooses ϕ_e to be those ϕ_{ij} for which i is even. The matrix A_4 is then block diagonal with each block a tridiagonal. The inversion of such a matrix is just a series of one-dimensional problems, which is what makes it possible to compute f_c . In this case the matrix A_c does not have the same form as A_f but may be factored into a series of one-dimensional problems. This approach is severely limited by the requirement that A_c factor exactly.

On the other hand conventional multigrid, guided by physical intuition and a desire to reduce the number of unknowns faster, defines ϕ_e as those ϕ_{ij} for which i and j are both even. For this choice all that can be said about A_4 in general is that it is wide banded and without any convenient structure. An example will be shown in the next section. The matrix A_c , computed using equation (8a), is much more difficult to solve than A_f and does not have the same form. Some sort of approximation seems in order.

One such approximation involves the standard decomposition

$$A_4 = D + (L + U) \quad (14)$$

where $D, L,$ and U are diagonal, lower triangular, and upper triangular matrixes respectively. Using this definition (5), (6), and (7) become

$$A_3 \phi_e + [D + (L + U)] \phi_o = f_o \quad (15)$$

$$\phi_o = D^{-1} [f_o - A_3 \phi_e - (L + U) \phi_o] \quad (16)$$

$$(A_1 - A_2 D^{-1} A_3) \phi_e = f_e - A_2 D^{-1} f_o + A_2 D^{-1} (L + U) \phi_o \quad (17)$$

respectively

It is often possible to express $A_2 D^{-1} (L + U) \phi_o$ in terms of ϕ_e and f_o . That is

$$A_2 D^{-1} (L + U) \phi_o = G \phi_e + H f_o + \epsilon \quad (18)$$

where G and H are matrices which are chosen to minimize the error term and simplify A_c . The approximation is necessary for it allows us to eliminate ϕ_o from equation (17). It is made possible by the fact that f_e , ϕ_e , and ϕ_o are related by a differential equation as well as by difference equations. Substituting equation (18) into equation (17) gives

$$(A_1 - A_2 D^{-1} A_3 - G) \phi_e = f_e - (A_2 D^{-1} - H) f_o + \epsilon \quad (19)$$

As a convenience we may left multiply equation (19) by an arbitrary diagonal matrix \hat{D} . In this case the two-dimensional equivalents of equations (8) are

$$A_c = \hat{D} (A_1 - A_2 D^{-1} A_3 - G) \quad (20a)$$

$$\phi_c = \phi_e \quad (20b)$$

$$f_c = \hat{D} [f_e - (A_2 D^{-1} - H) f_o + \epsilon] \quad (20c)$$

Derivation of the matrices G , H , and \hat{D} as well as the size of the resulting error term, ϵ , depends on the particular problem and boundary conditions. The coarse to fine grid interpolation involves approximations to the differential equation and is problem dependent. The nature of these approximations is best illustrated by an example. Such an example, that of a two dimensional Poisson problem is given here

5 Two Dimensional Example

As a specific two-dimensional example let us take

$$\phi_{xx} + \phi_{yy} = f(x, y) \quad (21a)$$

with boundary conditions

$$\phi(0, y) = f_1(y) \quad (21b)$$

$$\phi_x(\mathcal{L}, y) = f_2(y) \quad (21c)$$

$$\phi_v(x, 0) = f_3(x) \quad (21d)$$

$$\phi(x, \mathcal{L}) = f_4(x) \quad (21e)$$

discretized on a 4×4 equally spaced Cartesian grid so that $\Delta x = \Delta y = \frac{c}{4}$. Again we absorb the boundary data and a factor of Δx^2 into the right hand side. Also we adopt the double subscript notation where for example, $\phi_{(x_2, y_2)}$ is expressed as ϕ_{22} .

This discretization is

$$\left[\begin{array}{ccc|ccc|ccc|ccc}
 -4 & 1 & & & 1 & & & & & & & \\
 1 & -4 & 1 & & & 1 & & & & & & \\
 & 1 & -4 & 1 & & & 1 & & & & & \\
 & & 2 & -4 & & & & 1 & & & & \\
 \hline
 1 & & & & -4 & 1 & & 1 & & & & \\
 & 1 & & & 1 & -4 & 1 & & 1 & & & \\
 & & 1 & & & 1 & -4 & 1 & & 1 & & \\
 & & & 1 & & & 2 & -4 & & & 1 & \\
 \hline
 & & & & 1 & & & -4 & 1 & & 1 & \\
 & & & & & 1 & & 1 & -4 & 1 & & \\
 & & & & & & 1 & & -4 & 1 & & \\
 & & & & & & & 2 & -4 & & & 1 \\
 \hline
 & & & & & & & 2 & & & -4 & 1 \\
 & & & & & & & & 2 & & 1 & -4 \\
 & & & & & & & & & 2 & & \\
 & & & & & & & & & & 2 & -4
 \end{array} \right] \times \left[\begin{array}{c} \phi_{11} \\ \phi_{-1} \\ \phi_{-1} \\ \phi_{-1} \\ \phi_{12} \\ \phi_{-2} \\ \phi_{12} \\ \phi_{-2} \\ \phi_{13} \\ \phi_{-3} \\ \phi_{-3} \\ \phi_{14} \\ \phi_{-4} \\ \phi_{14} \\ \phi_{-4} \end{array} \right] = \left[\begin{array}{c} f_{11} \\ f_{21} \\ f_{-1} \\ f_{-1} \\ f_{12} \\ f_{22} \\ f_{-2} \\ f_{-2} \\ f_1 \\ f_{-1} \\ f_{-1} \\ f_4 \\ f_{-4} \\ f_{14} \\ f_{-4} \end{array} \right] \quad (22)$$

We permute this, choosing ϕ_e to be those $\phi_{i,j}$ for which i and j are both even

$$\left[\begin{array}{ccc|cccc} -4 & & & 1 & 1 & 1 & 1 \\ & -4 & & & 1 & 2 & \\ & & -4 & & & & 1 \\ & & & & & 2 & 1 \\ & & & & & & 2 \\ & & & & & & 2 \end{array} \right] \times \begin{bmatrix} \phi_{22} \\ \phi_{42} \\ \phi_{24} \\ \phi_{44} \\ \phi_{11} \\ \phi_{21} \\ \phi_{31} \\ \phi_{41} \\ \phi_{12} \\ \phi_{32} \\ \phi_{13} \\ \phi_{23} \\ \phi_{33} \\ \phi_{43} \\ \phi_{14} \\ \phi_{34} \end{bmatrix} = \begin{bmatrix} f_{22} \\ f_{42} \\ f_{24} \\ f_{44} \\ f_{11} \\ f_{21} \\ f_{31} \\ f_{41} \\ f_{12} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \\ f_{43} \\ f_{14} \\ f_{34} \end{bmatrix} \quad (23)$$

Consequently equation (17) becomes

$$\left[\begin{array}{cc|cc} -3 & \frac{1}{4} & \frac{1}{4} & \\ \frac{1}{2} & -3 & & \frac{1}{4} \\ \hline \frac{1}{2} & & -3 & \frac{1}{4} \\ & \frac{1}{2} & \frac{1}{2} & -3 \end{array} \right] \times \begin{bmatrix} \phi_{22} \\ \phi_{42} \\ \phi_{24} \\ \phi_{44} \end{bmatrix} = \begin{bmatrix} f_{22} \\ f_{42} \\ f_{24} \\ f_{44} \end{bmatrix} + \frac{1}{4} \begin{bmatrix} f_{21} + f_{12} + f_{23} + f_{32} \\ f_{41} + 2f_{32} + f_{43} \\ f_{14} + 2f_{23} + f_{34} \\ 2f_{34} + 2f_{43} \end{bmatrix} + A_2 D^{-1}(L + U)\phi_o \quad (24)$$

where the term $A_2 D^{-1}(L + U)\phi_o$ is given by

$$A_2 D^{-1}(L + U)\phi_o = \begin{bmatrix} -\frac{1}{2} & (\phi_{11} + \phi_{31} + \phi_{13} + \phi_{33}) \\ - & (\quad + \phi_{31} \quad + \phi_{33}) \\ - & (\quad \quad + \phi_{13} + \phi_{33}) \\ - & (\quad \quad \quad + 2\phi_{33}) \end{bmatrix} \quad (25)$$

We must relate this term to ϕ_e and f_e , using the fact that ϕ_e is not independent of ϕ_o . The two are related both through the difference equations and through the underlying differential equation (21). Use of the former would lead to an exact solution but would not

be much cheaper than solving equation (22) directly. Using the latter allows us to represent each line of equation (25) as a local discretization of equation (21) that is different from that used in equation (22). While this is cheap enough it will involve an approximation due to different truncation errors between the two discretizations of equation (21). We will show that it is possible to make the approximation a very good one by means of an appropriate relaxation process. This is the nature of multigrid methods.

Looking at the first line of equation (25) we have

$$-\frac{1}{2}(\phi_{11} + \phi_{31} + \phi_{13} + \phi_{33}) \quad (26)$$

Since we must use the differential equations to make our approximation we must know the physical location of these points in the domain. We find them to be the four diagonal neighbors of the point ϕ_{22} . If we expand each of them in x and y derivatives of ϕ about the point ϕ_{22} we find

$$-\frac{1}{2}(\phi_{11} + \phi_{31} + \phi_{13} + \phi_{33}) \simeq -2\phi_{22} - \left[\Delta x^2 \nabla^2 \phi + \frac{\Delta x^4}{12}(\phi_{xxxx} + 6\phi_{xxyy} + \phi_{yyyy}) \right]_{22} \quad (27)$$

The numerical Laplacian f_{22} may be related to the physical Laplacian $\Delta x^2 \nabla^2 \phi$ at the point (x_2, y_2) using the Taylor series analysis used in equation (26). This gives

$$f_{22} \simeq \left[\Delta x^2 \nabla^2 \phi + \frac{\Delta x^4}{12}(\phi_{xxxx} + \phi_{yyyy}) \right]_{22} \quad (28)$$

Adding equation (28) to equation (27) gives

$$-\frac{1}{2}(\phi_{11} + \phi_{31} + \phi_{13} + \phi_{33}) = -\phi_{22} - f_{22} - \frac{\Delta x^4}{2}(\phi_{xxyy})_{22} \quad (29)$$

One can make a similar argument for each line of equation (25) (see figure 1)

We see that G and H are now fully determined since no terms involving ϕ_o remain. The matrix \hat{D} is then determined from the constraint that A_c represents the same discretization of the problem as A_f but on a coarser mesh. In this case it is just $4I$.

For this example equation (19), after multiplication by \hat{D} , is given by

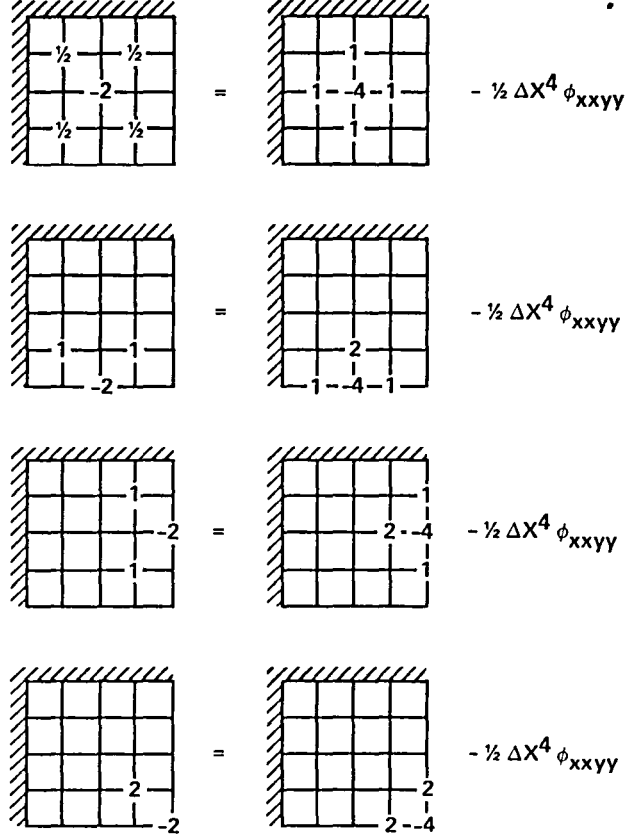


Figure 1 - Equivalent differencings
used in approximating $A_2 D^{-1}(L + U)\phi_0$

$$\left[\begin{array}{cc|cc} -4 & 1 & & 1 \\ 2 & -4 & & 1 \\ \hline 2 & & -4 & 1 \\ & 2 & 2 & -4 \end{array} \right] \times \begin{bmatrix} \phi_{22} \\ \phi_{42} \\ \phi_{24} \\ \phi_{44} \end{bmatrix} = \begin{bmatrix} f_{12} + f_{21} + f_{23} + f_{32} \\ f_{41} + 2f_{32} + f_{43} \\ f_{14} + 2f_{23} + f_{34} \\ 2f_{34} + 2f_{43} \end{bmatrix} + 2\Delta x^4 \phi_{xxyy} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (30)$$

which is indeed what we would like. Notice that a restriction operator has been suggested by the mathematics. This was arrived at by using all of the available information about the differential equations and the boundary conditions. It is applicable at the boundaries as well as in the center of the grid. Also of interest is that the expression for the error term is the same at the boundaries as it is in the interior of the grid. This turns out to be very helpful.

We now wish to recover ϕ_o from the now known ϕ_e . This may be done using the difference equations and an approximation. If, for example, we wish to find ϕ_{11} we may use the now known ϕ_{22} , the boundary conditions, and the interpolation formula given in equation (29) centered around the point (1,1) instead of the point (2,2). Similarly, all of the unknowns with two odd subscripts may be found by this formula. The remaining unknowns may now be found directly from the initial difference equations. This interpolation strategy, using the original difference equations as much as possible, is characteristic of MGR methods in general.

6 Smoothing

Ideally the error term $2\Delta x^4 \phi_{xxyy}$ in equation (30) would be zero. In practice it seldom is. In general there is very little that can be said *a priori* about this term. One escape from this dilemma is to use the correction formulation of equation (1). For this we add $A_f \phi_f^t$ to both sides of the negative of equation (1)

$$A_f(\phi_f^t - \phi_f) = A_f \phi_f^t - f_f \quad (31)$$

$$A_f e_f^t = r_f^t \quad (32)$$

The vector ϕ_f^t is the current guess for ϕ_f . (The more conventional notation, ϕ_f^n , has not been used here since n is used elsewhere.) The quantity r_f^t is called the residual and may be formed explicitly from known quantities. Since $\phi_f = \phi_f^t - e_f^t$, solving for e_f^t will yield ϕ_f . Also since equation (32) has the form of equation (1) all of the analysis developed for equation (1) will apply to equation (32). We adopt the notation that ϕ_c is a subset of e_f^t and f_c comes from r_f^t , i.e. from equation (30),

$$(f_c)_{11} = (r_f^t)_{12} + (r_f^t)_{21} + (r_f^t)_{23} + (r_f^t)_{32} \quad (32a)$$

This notational convenience frees us from having to refer to the “error of the error” as the grids become successively coarser.

It is possible to solve equation (1) by means of the nonstationary Point-Jacobi relaxation scheme (ref 5)

$$\phi^{t+1} = \phi^t - h^t(f - A\phi^t) \quad (33)$$

where h is a scalar iteration parameter. For classical Point-Jacobi $h = 1/4$

In this case it can be shown (ref 5), that the exact solution to equation (32) as a function of space and iteration number may be written as the double sum

$$e^t = \sum_{m=1}^M \sum_{n=1}^N \prod_{r=1}^t (1 + h^r \lambda_{mn}) c_{mn} X_{mn} \quad (34)$$

where for this problem (Neumann boundary conditions)

$$\lambda_{mn} = -4 + 2 \cos \frac{(m - \frac{1}{2})\pi}{M} + 2 \cos \frac{(n - \frac{1}{2})\pi}{N} \quad (34a)$$

$$(X_{mn})_{ij} = \sin \frac{i(m - \frac{1}{2})\pi}{M} \sin \frac{j(n - \frac{1}{2})\pi}{N} \quad (34b)$$

where as before i and j are space coordinates and t is the iteration number. The coefficients c_{mn} are determined from the initial guess. This is simply a decomposition of the error into the eigenvectors of A_f .

For this example the domain is a square of side \mathcal{L} . Using the identities $\mathcal{L} = M\Delta x = N\Delta y$, $x = i\Delta x$, and $y = j\Delta y$ equation (34b) may be written

$$X_{mn} = \sin \frac{(m - \frac{1}{2})\pi x}{\mathcal{L}} \sin \frac{(n - \frac{1}{2})\pi y}{\mathcal{L}} \quad (34c)$$

We define the attenuation factor σ_{mn} as

$$\sigma_{mn} = \prod_{r=1}^t (1 + h^r \lambda_{mn}) \quad (35)$$

We may now differentiate equation (34) directly to evaluate the error term of equation (30) when the process is applied to the correction equation (32). These are the same for each line of equation (32), namely

$$2\Delta x^4 \epsilon_{xxyy} = 2 \left(\frac{\mathcal{L}}{M}\right)^2 \left(\frac{\mathcal{L}}{N}\right)^2 \sum_{n=1}^N \sum_{m=1}^M \sigma_{mn} c_{mn} \left(m - \frac{1}{2}\right)^2 \left(n - \frac{1}{2}\right)^2 \left(\frac{\pi}{\mathcal{L}}\right)^4 X_{mn} \quad (36)$$

which we refer to collectively as the error in f_c . We define the error in an element of ϕ_c as the difference between that element and the corresponding element of ϕ_f . We may see the relation between the errors in f_c and those in ϕ_c by looking at one eigenvector at a time. The error in f_c due to X_{11} is

$$\frac{1}{8} \left(\frac{\pi}{M} \right)^2 \left(\frac{\pi}{N} \right)^2 c_{11} \sigma_{11} X_{11} \quad (37)$$

We observe that the error in f_c is exactly an eigenvector of A_f interpolated to the coarse mesh. It is also an eigenvector of A_c . We may therefore divide it by the corresponding coarse-grid eigenvalue to find the error in ϕ_c due to X_{11} . This eigenvalue is exactly

$$(\lambda_c)_{11} = -4 + 2 \cos \frac{(1 - \frac{1}{2})2\pi}{N} + 2 \cos \frac{(1 - \frac{1}{2})2\pi}{M} \quad (38)$$

Expanding the cosine terms about zero with a Taylor series and ignoring higher order terms gives

$$(\lambda_c)_{11} = -((\pi/M)^2 + (\pi/N)^2) \quad (39)$$

For this example $M = N$. Dividing equation (37) by equation (39) gives the error in ϕ_c due to X_{11} as

$$-\frac{1}{16} \left(\frac{\pi}{N} \right)^2 \sigma_{11} c_{11} X_{11} \quad (40)$$

Thus the eigenvector X_{11} is transferred to the coarse grid with second order accuracy. In a similar manner one can show that for $n \leq (N/2) - 1$, $m \leq (M/2) - 1$ the error in ϕ_c due to the eigenvector X_{mn} is

$$\frac{2(\pi/M)^2(\pi/N)^2[n - (1/2)]^2[m - (1/2)]^2 \sigma_{mn} c_{mn} X_{mn}}{-4 + 2 \cos[(2n - 1)\pi/N] + 2 \cos[(2m - 1)\pi/M]} \quad (41)$$

Of this group the worst case is when $n = (N/2) - 1$, $m = (M/2) - 1$. If M and N are large this error approaches

$$-\frac{\pi^4}{64} \sigma_{mn} c_{mn} X_{mn} \quad (42)$$

The eigenvector X_{mn} shows the shape of the error. The coefficient c_{mn} depends on the initial guess. The factor $-\pi^4/64$ is roughly -1.5 . Therefore, the attenuation factor σ_{mn} had better be less than $64/\pi^4$ in absolute value for this mode if it is to damp. This is easily done. As we will see in the next section, a much smaller value of σ is required for certain other modes because of aliasing.

7 Aliasing

While there are MN fine grid eigenvectors there are only $MN/4$ coarse grid eigenvectors. Therefore some of the fine grid eigenvectors are not accurately represented on the coarse grid. When $m > M/2$, or $n > N/2$ or both we find that there is no eigenvector on the coarse grid corresponding to X_{mn} on the fine grid. The error e_{xxyy} is not free from these components however, and when brought to the coarse grid, they appear as linear combinations of the coarse-grid eigenvectors. More specifically, the error appears as the eigenvector $X_{m'n'}$ on the coarse grid where

$$m' = \begin{cases} m & \text{for } m \leq M/2 \\ M + 1 - m & \text{for } m > M/2, \end{cases} \quad n' = \begin{cases} n & \text{for } n \leq N/2 \\ N + 1 - n & \text{for } n > N/2 \end{cases} \quad (43)$$

We now attempt to tailor σ_{mn} in such a way as to make σ_{mn} small for all values of m and n where X_{mn} makes a large contribution to errors in ϕ_c . Recall that

$$\lambda_{mn} = -4 + 2 \cos \frac{(m - \frac{1}{2})\pi}{M} + 2 \cos \frac{(n - \frac{1}{2})\pi}{N} \quad (34a)$$

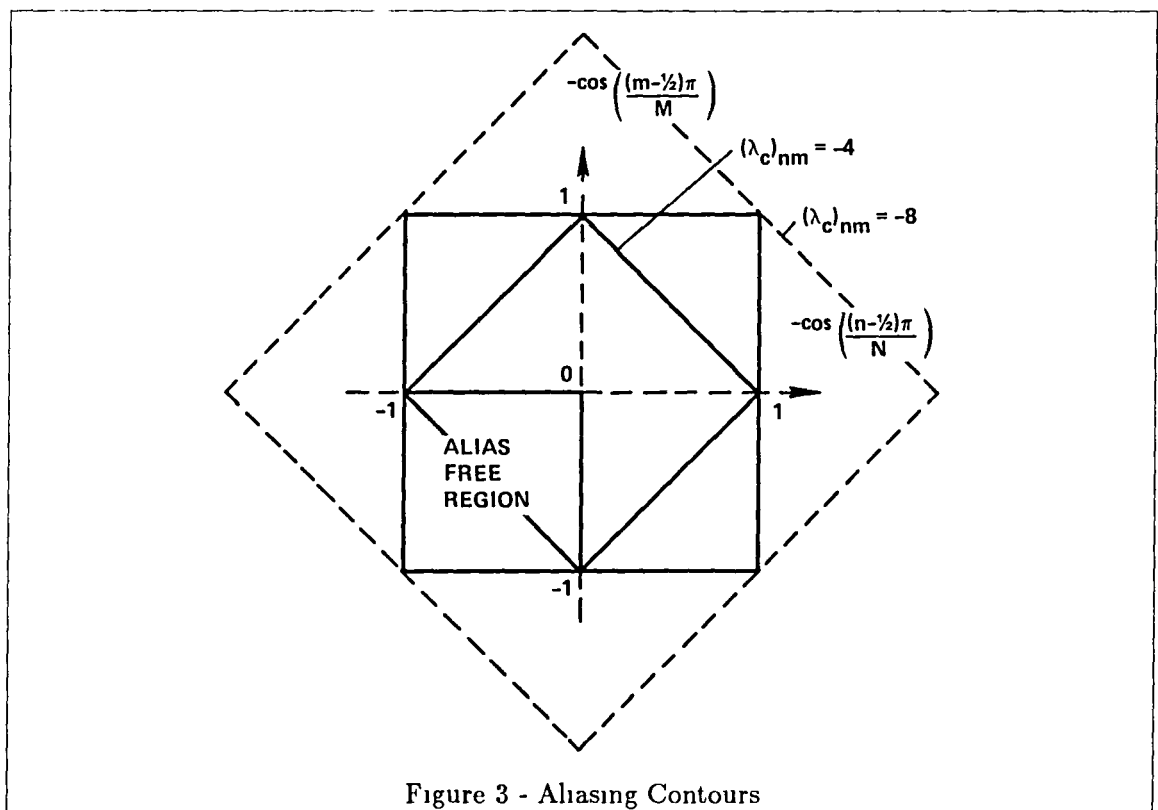
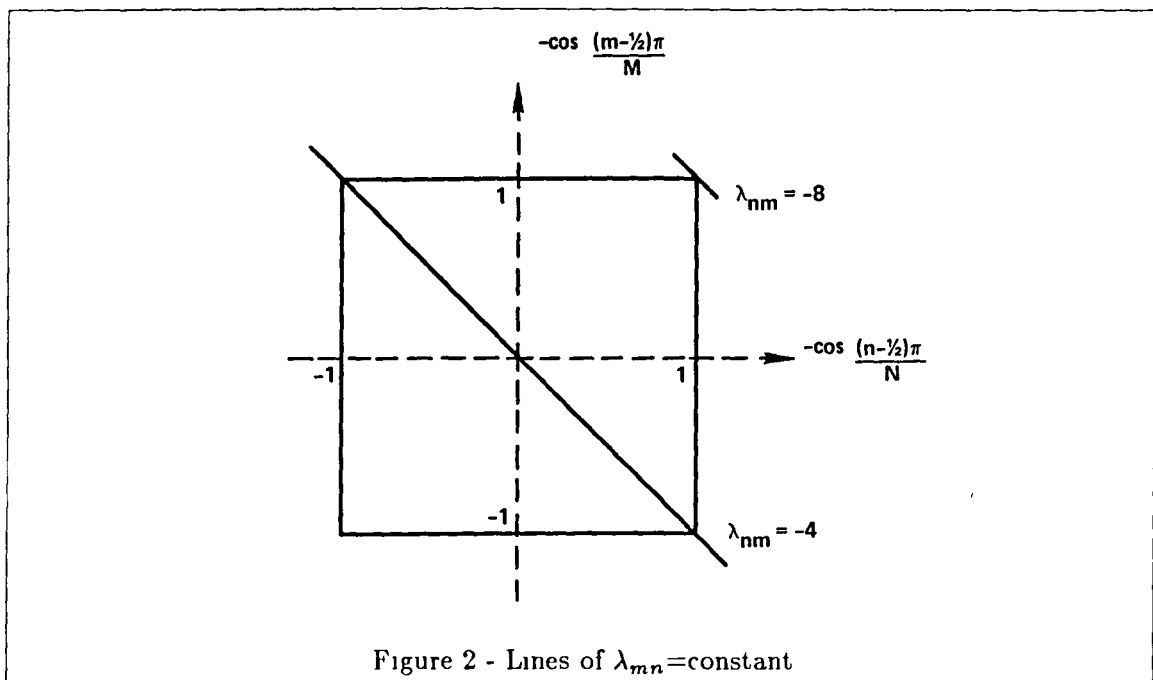
and

$$\sigma_{mn} = \prod_{r=1}^t (1 + h^r \lambda_{mn}) \quad (35)$$

The parameters that determine σ_{mn} are h^r . For the moment, let us allow λ_{mn} to have any value allowed by the range of the cosine terms. This gives a two-dimensional space of λ_{mn} which may be plotted as a square of side 2 centered at the origin. The actual boundaries of the square are not included in the region. (See figure 2)

We see that on this diagram lines of constant λ_{mn} have a slope of -1. The attenuation factor σ_{mn} will be constant along such lines as is evident from equation (35). Furthermore σ_{mn} will be zero when $h^r = -1/\lambda_{mn}$. Any value of $h < \frac{1}{4}$ is strongly stable in the sense that $|\sigma_{mn}| < 1$ for all m, n even if all the h^r have this value. One can also represent aliasing on this diagram by drawing contours of the fine grid λ_{mn} associated with eigenvectors which alias into coarse grid eigenvectors that all have the same value of λ_{mn} . These contours are diamond shaped as shown in figure 3.

Before making a parameter choice, we briefly review the two major sources of error. First there is the term $2\Delta x^4 e_{xxyy}$. Equation (36) shows that this term is largest for the eigenvector X_{MN} . In this case, the term $2\Delta x^4 e_{xxyy}$ for small Δx , would evaluate to,



$2\pi^4\sigma_{MNC_{MN}}X_{MN}$ The shape of this mode is preserved but the amplitude is off by about a factor of 200. Actually this is only half of the story

The second source of error is aliasing. The largest aliasing errors occur on eigenvectors which alias into the coarse-grid eigenvector associated with the smallest coarse-grid eigenvalue. From equation (39) this is,

$$(\lambda_c) \simeq -((\pi/M)^2 + (\pi/N)^2) \quad (44)$$

Eigenvectors which alias into the eigenvector X_{11} on the coarse grid have their error term multiplied by the inverse of this quantity and deposited in X_{11} . The fine grid eigenvalues in question are X_{MN} , X_{M1} , X_{1N} , and X_{11} . (The fine grid eigenvector X_{11} is in fact what we would like to transfer to the coarse grid eigenvector X_{11} . It therefore can not be said to alias but is included here for comparison purposes.) Evaluating equation (41) by replacing m and n in the denominator with N' gives errors in ϕ_c of

$$-\pi^2 N^2 \sigma_{MNC_{MN}} X_{MN} \quad (44a)$$

$$-\frac{\pi^2}{4} \sigma_{M1} c_{M1} X_{M1} \quad (44b)$$

$$-\frac{\pi^2}{4} \sigma_{1N} c_{1N} X_{1N} \quad (44c)$$

$$-\left(\frac{\pi}{4N}\right)^2 \sigma_{11} c_{11} X_{11} \quad (44d)$$

respectively. The first of these is unbounded if σ_{MN} is 1. Clearly this can't be tolerated. We would like to pick h^τ such that σ_{MN} is proportional to N^{-4} . By choosing $h^{(1)} = h^{(2)} = \frac{1}{8}$ we are led to $\sigma_{MN} = \left(\frac{1}{64}\right)\left(\frac{\pi}{N}\right)^4$. Using equation (44a) this leads to an error on the coarse grid due to this mode of $(1/N^2)(\pi^6/64)$ which goes to zero in a second order way. Thus we have completely neutralized the threat of aliasing from X_{MN} . Errors from this source are of the same order as errors from X_{11} .

The errors represented by equation (44b) and equation (44c) must also be attenuated or they will dominate. We are guided by a desire to preserve the accuracy of the eigenvector X_{11} since this will be represented on even the coarsest mesh. The relaxation selected in the previous paragraph also works on eigenvectors X_{1M} and X_{N1} , but not as well. The attenuation factor for each step is $\sigma_{1M} = \sigma_{N1} = \frac{1}{2}$. Since there were two steps, the total attenuation for these terms is one fourth. The coefficients are then reduced from π^2 to $\pi^2/4$. We would like them to be proportional to N^{-2} so that errors in each of these modes decline in a second order way. To do this we select $h^{(3)} = \frac{1}{4}$. This results in σ_{N1} and σ_{1M} both being equal to zero.

Finally, we look back to the errors in the components that don't alias. At the end of section 6 these were determined to be greatest in the eigenvector $X_{M,N}$, which has a

corresponding eigenvalue approaching 4. This is the same eigenvalue as X_{M1} and X_{1N} . No further smoothing is required to reduce errors in components that don't alias.

The question of smoothing parameters can be resolved for the model problem. The coarse grid error component corresponding to the eigenvalue of smallest modulus is the one to protect. All modes which alias onto this mode should be attenuated until the resulting coarse grid error decreases with N . Their corresponding eigenvalues may often be estimated using Gerschgorin's theorem (Ref. 6). On problems which are within a perturbation of the model problem one might use the parameters given here scaled according to the largest eigenvalue.

For the three-step relaxation just discussed, $\sigma_{mn} = [1 + (\lambda_{mn}/8)]^2 [1 + (\lambda_{mn}/4)]$, which is plotted in figure 4. Thus, all of the eigenvectors which alias are severely reduced in amplitude and those which alias most are reduced most. The eigenvectors which don't alias are also reduced in amplitude.

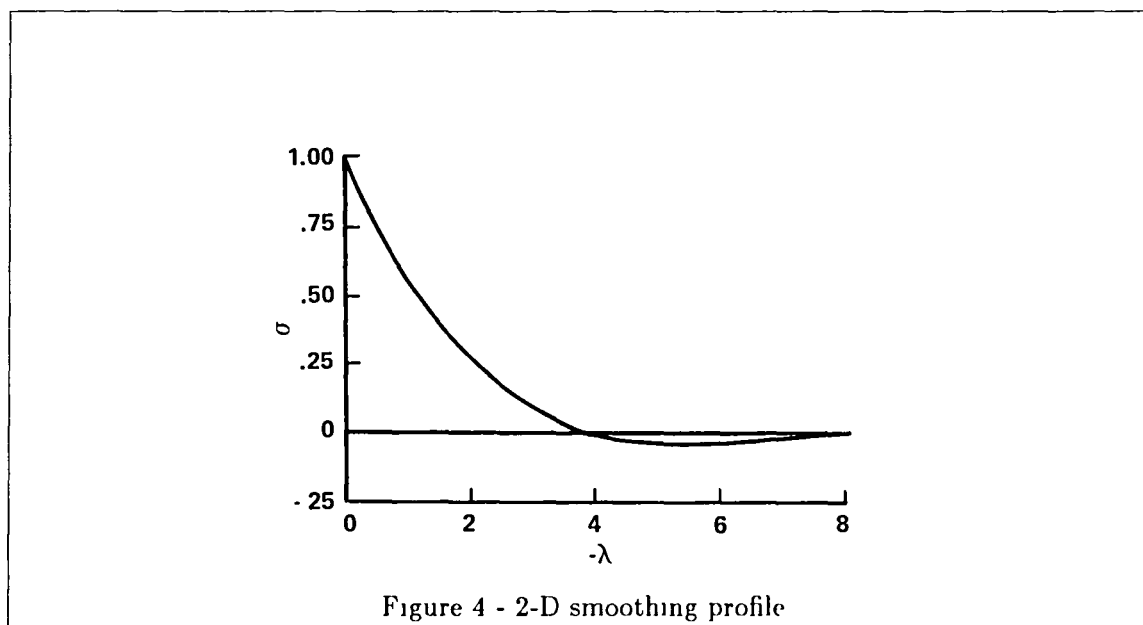


Figure 4 - 2-D smoothing profile

8 Coarse To Fine Interpolation

The errors incurred during the interpolation pose less of a problem than those incurred during the restriction process. This is because the relevant error terms appear in the unknowns rather than in the right hand side. Furthermore, aliasing does not occur since all the coarse grid eigenvectors are representable on the fine grid.

In what follows, we will explore and analyze one possible interpolation strategy. This strategy is motivated by the same geometrical arguments used in forming the restriction operator.

$$(1) \quad (e_f)_{ij} = (\phi_c)_{\frac{i}{2}, \frac{j}{2}}$$

if i and j are even

$$(2) \quad (e_f)_{ij} = \frac{1}{4} [(e_f)_{i+1j+1} + (e_f)_{i-1j+1} \\ + (e_f)_{i+1j-1} + (e_f)_{i-1j-1} \\ - 2(r_f)_{ij}]$$

if i and j are odd

$$(3) \quad (e_f)_{ij} = \frac{1}{4} [(e_f)_{i+1j+1} + (e_f)_{ij+1} \\ + (e_f)_{i-1j} + (e_f)_{ij-1} \\ - (r_f)_{ij}]$$

for all i and j . This comes directly from the original difference equations for these points

- (4) Improve the estimate of ϕ_f by subtracting e_f

An explanation of the above strategy follows. In step 1 we simply assign coarse grid values to the fine grid at those points where the two grids coincide. In step 2, we again use the rotated difference equations used during the derivation of the restriction operator. This carries with it a fourth order error term which causes inaccuracies in all of the eigenvector components of e_f . In step 3, we use the difference equations to fill in all the missing values. Finally, in step 4, we use our knowledge of the fine grid error to improve the fine grid solution. Optionally one can do some more smoothing to remove the errors incurred during this interpolation process. Though this improves the convergence per step it was found not to be cost effective. See the discussion of operation count for more details.

9 A Three-Dimensional Example

The example chosen in three dimensions is again Poisson's equation using purely Dirichlet boundary conditions. Again we use an equally spaced Cartesian mesh and the standard seven-point differencing star. The domain is a cube of side \mathcal{L} . The analysis is exactly the same as for the two-dimensional problem although the matrices are much larger. Only the results of the analysis will be given here.

We are led to the following approximation in arriving at the restriction operator

$$\begin{aligned}
& \frac{1}{4}(\phi_{i+1,j,k+1} + \phi_{i-1,j,k+1} + \phi_{i+1,j,k-1} + \phi_{i-1,j,k-1} \\
& + \phi_{i,j+1,k+1} + \phi_{i,j+1,k-1} + \phi_{i,j-1,k+1} + \phi_{i,j-1,k-1} \\
& + \phi_{i+1,j+1,k} + \phi_{i-1,j+1,k} + \phi_{i+1,j-1,k} + \phi_{i-1,j-1,k}) \\
& = 3\phi_{i,j,k} + f_{i,j,k} + \frac{\Delta x^4}{4}(\phi_{xxyy} + \phi_{yyzz} + \phi_{zzxx})
\end{aligned}$$

This yields the restriction operator

$$\begin{aligned}
(f_c)_{i,j,k} &= (r_f)_{2i+1,2j,2k} + (r_f)_{2i-1,2j,2k} + (r_f)_{2i,2j+1,2k} \\
&+ (r_f)_{2i,2j-1,2k} + (r_f)_{2i,2j,2k+1} + (r_f)_{2i,2j,2k-1} - 2(r_f)_{2i,2j,2k}
\end{aligned}$$

In analyzing the smoothing, the three-dimensional analog of figure 2 is a cube instead of a square. Reasoning, as in section 7, we choose four relaxation sweeps with values of $h^r = \frac{1}{12}$, $\frac{1}{12}$, $\frac{1}{8}$, and $\frac{1}{4}$. The last is not strongly stable but the sequence is stable. Notice the similarity to the two-dimensional case where the worst error is smoothed twice and the other errors are smoothed once.

The coarse to fine-grid interpolation is similar to that for two dimensions although one more approximation is required

$$1) \quad (e_f)_{i,j,k} = (\phi_c)_{\frac{i}{2}, \frac{j}{2}, \frac{k}{2}}$$

for i, j, k , even

$$\begin{aligned}
(2) \quad (e_f)_{i,j,k} &= \frac{1}{8} [(e_f)_{i+1,j+1,k+1} + (e_f)_{i+1,j+1,k-1} \\
&+ (e_f)_{i+1,j-1,k+1} + (e_f)_{i+1,j-1,k-1} \\
&+ (e_f)_{i-1,j+1,k+1} + (e_f)_{i-1,j+1,k-1} \\
&+ (e_f)_{i-1,j-1,k+1} + (e_f)_{i-1,j-1,k-1} \\
&- 4(r_f)_{i,j,k}]
\end{aligned}$$

if i, j , and k are odd

$$\begin{aligned}
(3) \quad (e_f)_{i,j,k} &= \frac{1}{4} [(e_f)_{i+1,j,k} + (e_f)_{i-1,j,k}] \\
&+ \frac{1}{8} [(e_f)_{i,j+1,k+1} + (e_f)_{i,j+1,k-1} \\
&+ (e_f)_{i,j-1,k+1} + (e_f)_{i,j-1,k-1}]
\end{aligned}$$

$$-\frac{1}{4} (r_f)_{ijk}$$

for i odd, j and k even and for k odd, i , and j even

$$(4) \quad (e_f)_{ijk} = \frac{1}{6} [(e_f)_{i+1jk} + (e_f)_{i-1jk} \\ + (e_f)_{ij+1k} + (e_f)_{ij-1k} \\ + (e_f)_{ijk+1} + (e_f)_{ijk-1} \\ - (r_f)_{ijk}]$$

for all indices This is just using the original difference equations

- 5) Improve the estimate of ϕ_f by subtracting e_f

10 Summary

The process just described is summarized as follows

- 1) Smoothing to reduce the error incurred during restriction This is done using nonstationary Point Jacobi relaxation with the h^t selected above
- 2) Computation of the required fine grid residuals
- 3) Transferring the problem to a coarser mesh using the restriction operator derived above
- 4) Exact solution of the problem on the coarser mesh If the coarsest mesh has more than 1 unknown, "exact" solution may be the result of some suitable relaxation process This will be cheap since the coarsest mesh has very few unknowns On other than the coarsest mesh "exact" solution means two iterations of this multigrid process (This is the so called W-cycle)
- 5) Transferring the solution back to the fine mesh using the coarse to fine interpolation given above
- 6) Repetition of steps 1 through 5 until convergence is obtained There will be further discussion of what is meant by convergence

11 Operation Count

In this section we address the total cost of ACR In two dimensions, a nonstationary Point Jacobi relaxation for the five point Laplacian requires 7 operations per point where multiplications and additions are both counted Interpolations account for about 30% of the

Table 1 ACR Operation Count		
2-D	Description Of Process Segment	3-D
21	finest mesh smoothing	36
3	computation of residuals	4
1	restriction operator	$\frac{7}{8}$
4	interpolation operator	$6\frac{7}{8}$
2^4	total for finest mesh	$47\frac{7}{8}$
$\times 2$	factor for W-cycle	$\times \frac{4}{3}$
58	total for all meshes	$63\frac{2}{3}$

total For three dimensions the relaxation sweeps require 9 operations/point Interpolations require only about 25% of the total The operation count for both is given in table 1

In both of the above operation counts we have taken into account the fact that restriction only occurs at fine mesh points with all even subscripts This means that we do not need the residuals everywhere The count for both of these reflects the fact that they are not done at every point The factor for the W-cycle assumes an infinite number of grids In the two-dimensional case for example, each grid requires one-fourth the number of operations of the next finer grid, but must be visited twice for each time the finer grid is visited This leads to the series

$$1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots = 2$$

which is where that factor of two comes from In three dimensions each grid requires only one-eighth the number of operations of the next finer grid This leads to a factor of four thirds Notice, that because the number of operations on coarse grids is proportionately less in three dimensions than in two dimensions, the cost of an additional relaxation sweep is also less even though the difference stencil is larger

This scheme was devised for ease of explanation rather than for speed Possible speed improvements include

- 1) Improved relaxation schemes such as checkerboard Gauss-Seidel or incomplete *LU* decomposition These schemes are more efficient at removing all the restriction errors and require no parameter choice, but are more difficult to analyze
- 2) Configuring the scheme as an FMG cycle (ref 1) In this case, the method would start with an exact solution on the coarsest mesh It would then proceed as described above but starting at the point where the coarsest grid exact solution is computed This can be thought of as producing a better initial guess on the finest grid at minimal cost Some investigators have found that only one additional cycle is required to reduce the errors to the level allowed by our finite difference approximation It rarely makes sense to

reduce the errors to a still lower level. All the cases run here were converged to machine accuracy however, since our intent was to test the convergence properties of the method.

- 3) Visiting each mesh only once instead of twice. This is the so called V-cycle. This gives a savings of one-third in the operation count per step (one-seventh in 3-D) but weakens the bounds on the spectral norm of the method because the 'exact' solution on intermediate grids is not as good. Consequently more steps may be required. There is some practical experience to the contrary (ref. 7).

12 Invariant Subspace Analysis

The question of errors introduced during the interpolation was only touched on briefly. In fact it is these errors that allow reintroduction of high-frequency error on the finest grid. Without them the troublesome components would soon disappear and the restriction would become nearly exact. With exact interpolation and restriction multigrid becomes a direct method. Since these errors limit convergence, it is necessary to take them into account when analyzing multigrid methods. The best (perhaps only) quantitative analysis of interpolation errors for the model problem is the method of invariant subspaces. This is explained in some detail in (ref. 7).

In the section on restriction it was shown that in two dimensions the four fine grid eigenvectors X_{mn} , $X_{m'n}$, $X_{mn'}$, and $X_{m'n'}$ all appear on the coarse grid as the X_{mn} eigenvector. In the section on interpolation we briefly outlined a way of treating errors that only occur on certain points. If we pursue this, we find that when the coarse grid eigenvector X_{mn} is interpolated to the fine grid, errors are introduced in only the four eigenvectors just mentioned. This nice property is preserved through the smoothing, restriction, and computation of residuals as well. Thus the error in these four components at the end of a step depends only on their errors at the beginning of that step. We can analytically form the 4×4 matrix which represents this situation. To find the error at the end of T steps we simply multiply the initial error by the T th power of this matrix. The spectral radius of the method is just the largest of the spectral radii of these 4×4 matrices and the spectral norm is the largest of their spectral norms. Using this method we have numerically computed these quantities for the methods advocated here. Also we computed these quantities for different amounts of smoothing. Adding a smoothing sweep can decrease the spectral radius and norm but will increase the cost. A function which follows this tradeoff is

$$F = - \frac{\log \text{ of spectral norm}}{\text{number of operations}}$$

Loosely, this is the number of base e digits per multiply.

In tables 2 and 3 we show the performance of ACR with different amounts of smoothing. For completeness we include the possibility of post interpolation smoothing. The numbers are limiting values as $\Delta x \rightarrow 0$.

The three-dimensional case is completely analogous although the invariant subspaces each contain eight components instead of four. Consequently there are three eigenvalues cor-

Table2 2-D Convergence Results								
Smoothing						Spectral Radius	Spectral Norm	F
t	Pre-Restriction			Post-Interp				
0						1 255	∞	$-\infty$
1	$\frac{1}{8}$					0 500	1 250	-0 07
2	$\frac{1}{8}$	$\frac{1}{8}$				0 249	0 390	0 21
3	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{4}$			0 114	0 118	0 37
4	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{6}$		0 074	0 092	0 33
5	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{2}$	0 072	0 073	0 30

Table 3 3-D Convergence Results									
Smoothing							Spectral Radius	Spectral Norm	F
t	Pre-Restriction				Post-Interp				
0							3 323	∞	$-\infty$
1	$\frac{1}{12}$						1 188	∞	$-\infty$
2	$\frac{1}{12}$	$\frac{1}{12}$					0 297	1 001	-0 000
3	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{8}$				0 220	0 362	0 017
4	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{8}$	$\frac{1}{4}$			0 148	0 192	0 023
5	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{6}$		0 116	0 160	0 022
6	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{3}$	0 089	0 132	0 021

responding to error components aliasing into the smallest eigenvalue instead of two (and seven eigenvectors instead of three)

13 Results

The 2-D results are for the example in section 5. Eight mesh sizes varying between $N = 2$ and $N = 256$ were tried. All the test cases were reducible to one unknown. We chose the homogeneous case where $f_f = \phi_f = 0$. This was chosen to simplify computation of the error (which for this case is just the current estimate for ϕ_f) and does not imply that the process is restricted to homogeneous boundary conditions (ref 8). The initial guess was chosen so that all the coefficients c_{mn} were equal and of such a magnitude as to make the L_2 norm of the error equal to 1. The complete convergence history is given in table 4, for a 256×256 grid using the three step smoothing suggested in section 7.

Each complete cycle reduced the L_2 norm of the error by a factor of 27 or more. The spectral norm guarantees a factor of 8.5 per step but this is overly pessimistic. Like any linear iterative scheme, ACR starts out fast and then slows down to some convergence rate which depends on the spectral norm or radius. Its advantage is twofold, first, the asymptotic rate is independent of N , quicker than any other explicit method and second, the problem may well be converged before this limit is reached. The independence of the spectral norm on N has been shown for other multigrid methods (ref 7). Our experience has shown no

Table 4 2-D Convergence History—256 × 256 Grid		
Step	$\ e\ _2$	$\ e\ _\infty$
0	1×10^0	5.3×10^4
1	1.3×10^{-2}	3.1×10^2
2	2.3×10^{-4}	6.9×10^0
3	2.6×10^{-6}	5.0×10^{-2}
4	6.3×10^{-8}	7.9×10^{-4}
5	2.3×10^{-9}	4.6×10^{-5}

degradation of convergence over a wide range of values for N

The 3-D results are for the example in section 9. Four mesh sizes varying between $N = 3$ and $N = 31$ were tried. All the test cases were reducible to one unknown. We again chose the homogeneous case where $f_f = \phi_f = 0$. The initial guess was chosen so that all the coefficients c_{mno} were equal and of such a magnitude as to make the L_2 norm of the error equal to 1. The complete 3-D convergence history is given in table 5 for a $31 \times 31 \times 31$ grid using the four step relaxation suggested in section 9.

Table 5 3D Convergence History-31 × 31 × 31 Grid		
Step	$\ e\ _2$	$\ e\ _\infty$
0	1.0×10^0	2.3×10^4
1	2.0×10^{-2}	3.9×10^2
2	6.6×10^{-4}	9.2×10^0
3	2.6×10^{-5}	3.4×10^{-1}
4	2.9×10^{-6}	1.6×10^{-2}
5	7.3×10^{-8}	8.5×10^{-4}

Each complete cycle reduced the L_2 norm of the error by a factor of 9 or more. The spectral norm guarantees a factor of 5.2 per step.

Thus we see that the remarkable results claimed by the analysis are actually realized in practice. No other type of explicit method allows an entire convergence history of this problem to be written down in a short table. Furthermore, 3-D problems take only about 1.5 times as much work per point as 2-D problems, an important feature of multigrid methods.

In comparing ACR with other explicit methods and with cyclic reduction I will use table 6 prepared by Dorr (ref. 9). This is for Poisson's equation discretized on a square with N^2 unknowns. The direct methods are compared with the iterative ones by assuming that a reduction of the error by a factor of N^2 is required. This comes from the fact that the truncation error is proportional to $\frac{1}{N^2}$. For purposes of comparison all acceleration parameters are optimally chosen.

The factor of $\log_2 N$ in ACR and MGR-CH_{2,1} does not come from any specific feature of the algorithm but from the fact that the desired accuracy increases with the number

Table 6 Method Comparison	
Method	Operation Count
Block (polynomial form)	$6N^2$
Block (Schechter)	$\frac{3}{2}N^3$
Block (Froehlich)	$(\frac{26}{3} + 4r)N^3$
Odd-even reduction (Buzbee et al)	$\frac{9}{2}N^2 \log_2 N$
Tensor product (Lynch et al)	$8N^3$
Fourier Series (Hockney)	$5N^2 \log_2 N$
SOR	$\frac{7}{2}N^2 \log_2 N$
ADI	$4N^2(\log_2 N)^2$
ACR	$\sim 38N^2 \log_2 N$
MGR-CH _{2,1}	$\sim 17N^2 \log_2 N$

of unknowns. Limited precision on a given computer may limit the attainable accuracy. Under such conditions the method requires order N^2 operations to achieve this limited precision. In any event, given the restrictions on memory size common in today's computers, $\log_2 N < 10$. In practice multigrid methods can be made to be of order N^2 by use of the FMG cycle outlined previously in reference 1 but the coefficients will increase from 38 and 17 to 96 and 50 for ACR and MGR-CH_{2,1} respectively. The FMG cycle is usually good if the initial guess is largely random. If, on the other hand there is a reasonable guess from some nearby problem the basic W-cycle will probably converge in one or two steps.

The improved performance of MGR-CH_{2,1} over ACR is due largely to its use of checkerboard Gauss-Seidel for the removal of high frequency error components. The operation count for this method is much less than for the non-stationary point-Jacobi relaxation used in ACR.

14 Conclusions

For the multigrid process just presented it is possible to formally analyze errors made in the interpolation and restriction processes on these model problems. It then becomes possible to tailor the smoothing according to these errors. The analysis yields interpolations and restrictions that are valid at Neumann boundaries as well as in the interior of the domain.

We have shown that cyclic reduction can be thought of as a particular multigrid method that has exact interpolation and restriction. This is particularly evident in one dimension where the two methods coincide. Although the two methods differ in higher dimensions, they are equivalent up to a known approximation. This approximation can be improved with an appropriate relaxation. Using the difference equations to do the interpolation eliminates the need for post-interpolation smoothing. An efficient explicit method results.

The real value of multigrid techniques comes from applications to problems which cannot be solved with cyclic reduction. Since ACR can be viewed in terms of point operators it may prove easier to adapt to complicated grid structures than cyclic reduction.

References

- 1 Brandt, Achi, "Multi-Level Adaptive Solutions to Boundary Value Problems," Mathematics of Computation, vol 31, No 138, April 1977, pp 333-390
- 2 Brown, Jeffrey, J, "A Multigrid Mesh Embedding Technique for Three Dimensional Transonic Potential Flow Analysis," Proceedings of a symposium on Multigrid Methods, Ames Research Center, Moffett Field, CA, October 21-22, 1981, pp 131-150
- 3 Jameson, Antony, "Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multiple Grid Method," AIAA Paper 79-1458
- 4 Dorr, Fred, "The Direct Solution of the discrete Poisson Equation on a Rectangle, SIAM Review, Vol 12, No 2, April 1970, pp 248-260
- 5 Forsythe, George and Wasow, Wolfgang, "Finite difference Methods for Partial Differential Equations," John Wiley and Sons, Inc , pp 226-235
- 6 Wilkinson, J H , "The Algebraic Eigenvalue Problem," Oxford University Press , p 71
- 7 Stuben, K and Trottenberg, U , "Multigrid Methods Fundamental Algorithms, Model Problem Analysis and Applications," Proceedings of Multigrid Methods conference at Koln-Porz , 1981 , p 50
- 8 Forsythe, George and Wasow, Wolfgang, "Finite Difference Methods for Partial Differential Equations," John Wiley and Sons, Inc , pp 214-215

1 Report No NASA TM-86020	2 Government Accession No	3 Recipient's Catalog No	
4 Title and Subtitle ON THE CONNECTION BETWEEN MULTIGRID AND CYCLIC REDUCTION		5 Report Date September 1984	
		6 Performing Organization Code	
7 Author(s) Marshal L. Merriam		8 Performing Organization Report No A-9884	
		10 Work Unit No T-6465	
9 Performing Organization Name and Address Ames Research Center Moffett Field, CA 94035		11 Contract or Grant No	
		13 Type of Report and Period Covered Technical Memorandum	
12 Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14 Sponsoring Agency Code 505-31-01	
15 Supplementary Notes Point of Contact: Marshal L. Merriam, Ames Research Center, MS 202-A1, Moffett Field, CA 94035 (415) 694-6417 or FTS 464-6417			
16 Abstract A technique is shown whereby it is possible to relate a particular multigrid process to cyclic reduction using purely mathematical arguments. This technique suggests methods for solving Poisson's equation in 1-, 2-, or 3-dimensions with Dirichlet or Neumann boundary conditions. In one dimension the method is exact and, in fact, reduces to cyclic reduction. This provides a valuable reference point for understanding multigrid techniques. The particular multigrid process analyzed is referred to here as Approximate Cyclic Reduction (ACR) and is one of a class known as Multigrid Reduction methods in the literature. It involves one approximation with a known error term. It is possible to relate the error term in this approximation with certain eigenvector components of the error. These are sharply reduced in amplitude by classical relaxation techniques. The approximation can thus be made a very good one.			
17 Key Words (Suggested by Author(s)) Multigrid methods Fast solvers Poisson equation		18 Distribution Statement Unlimited Subject category - 64	
19 Security Classif (of this report) Unclassified	20 Security Classif (of this page) Unclassified	21 No of Pages 28	22 Price* A03

End of Document